The RM documentation and existing plugins use the Rails `unloadable` statement. This doesn't work as expected with Redmine 3.1. I have quite a bit of experience with Rails and building RM plugins so please do not disregard this without an investigation.

The following steps illustrate and reproduce the problem:

1) Create a fresh new Redmine:

   git clone https://github.com/redmine/redmine.git
cd redmine
git checkout 3.1-stable

   nano config/database.yml
   bundle exec rake generate_secret_token
   bundle exec rake db:migrate

2) Create a fresh new plugin:

   rails g redmine_plugin polls
   rails g redmine_plugin_model polls poll name:string

   bundle exec rake redmine:plugins:migrate

3) Add any code reference to this new `unloadable` Poll model. A quick ways is to add the following to plugins/polls/init.rb:

   Rails.application.config.to_prepare do
     Poll.new
   end

4) At this point when Rails need to do code reloading it will fail with a message indicating it expected plugin/polls/app/models/poll.rb to define Poll. Try it by doing the following:

   - Run the rails server and open any page served by it in your browser.
   - Edit the any .rb file. An example is to add a newline to plugins/polls/app/models/poll.rb.
   - Click any link in the page opened in the first step. You will get: **Unable to autoload constant Poll, expected**
If you remove the `unloadable` in the plugin Poll model this problem disappears. Note that a very large amount of RM plugins currently use `unloadable` and causes this issue. My environment after performing these 4 steps:

```
Environment:
  Redmine version    3.1.0.stable
  Ruby version       2.1.3-p242 (2014-09-19) [x86_64-linux]
  Rails version      4.2.3
  Environment        production
  Database adapter   PostgreSQL
SCM:
  Subversion         1.8.8
  Git                1.9.1
  Filesystem
  Redmine plugins:
    polls            0.0.1
```

Related issues:
Related to Redmine - Patch # 27963: Remove `unloadable` from bundled sample p...
[Closed]

Associated revisions
Revision 17160 - 2018-01-08 18:53 - Jean-Philippe Lang

Removes 'unloadable' from plugin templates (#20513).

Patch by Go MAEDA.

Revision 17161 - 2018-01-08 18:54 - Jean-Philippe Lang

Merged r17160 to 3.4-stable (#20513).

History
#1 - 2015-08-11 18:35 - Pierre Pretorius

Note that I can reproduce this in standard Rails 4.2.3 very easily. I've logged it on github: [https://github.com/rails/rails/issues/21198](https://github.com/rails/rails/issues/21198)

1) Create new Rails app:

```
rails new polls
bundle exec rails generate scaffold Question name:string
bundle exec rake db:migrate
```

2) Make a model unloadable:

```
nano app/models/question.rb
```
3) Do something that causes Rails to reload code:

  rails s &
  curl localhost:3000/questions -s > /dev/null
  touch app/models/question.rb
  curl localhost:3000/questions -s > /dev/null

Error displayed: **Unable to autoload constant Question, expected app/models/question.rb to define it.**

#2 - 2015-10-04 11:05 - Jens Krämer

+1

Just stumbled across the very same problem and indeed calling unloadable on plugin models (or on core models when patching them) was the culprit.

In addition to that unloadable does not seem to serve any real purpose (at least nowadays) - Rails' auto loading just works like a charm for both core and plugin code without it.

#3 - 2016-11-16 19:52 - Javier Menéndez Rizo

When I quit the unloadable, I'm getting this from the core of redmine:

```
NameError
  Couldn't find CustomFieldsHelper, expected it to be defined in helpers/custom_fields_helper.rb
```

#4 - 2016-12-25 14:36 - Javier Menéndez Rizo

Javier Menéndez wrote:

```
When I quit the unloadable, I'm getting this from the core of redmine:

  NameError
   Couldn't find CustomFieldsHelper, expected it to be defined in helpers/custom_fields_helper.rb

[...]
```

I just removed all unloadable references from my plugins and the reload is working again.

This was on redmine 3.3.1 and ruby 2.3.1

#5 - 2017-07-01 02:26 - Akiko Takano

+1

I saw the same situation after Redmine3.x.

This may be wrong, but I personally think, at least the plugin which support version is Redmine3.x or higher, does not need to state unloadable.
Can we remove all 'unloadable' from plugin templates?

Index: lib/generators/redmine_plugin_controller/templates/controller.rb.erb

---
lib/generators/redmine_plugin_controller/templates/controller.rb.erb
(revision 16946)
+++ lib/generators/redmine_plugin_controller/templates/controller.rb.erb
(working copy)
@@ -1,6 +1,4 @@
class <%= @controller_class %>Controller < ApplicationController
- unloadable
-<% actions.each do |action| -%>
def <%= action %>
Index: lib/generators/redmine_plugin_model/templates/model.rb.erb

---
lib/generators/redmine_plugin_model/templates/model.rb.erb
(revision 16946)
+++ lib/generators/redmine_plugin_model/templates/model.rb.erb
(working copy)
@@ -1,3 +1,2 @@
class <%= @model_class %> < ActiveRecord::Base
- unloadable
end

#7 - 2017-12-02 11:28 - Akiko Takano

Go MAEDA wrote:

Can we remove all 'unloadable' from plugin templates?

[...]

Yes we can, I personally think so.

#8 - 2017-12-10 13:10 - Go MAEDA

- Target version set to 4.0.0

Setting target version to 4.0.0.

#9 - 2018-01-08 18:54 - Jean-Philippe Lang

- Status changed from New to Closed
- Target version changed from 4.0.0 to 3.4.4
- Resolution set to Fixed

Change committed, thanks.
- Related to Patch #27963: Remove ‘unloadable’ from bundled sample plugin added